

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions and listings of claims in the application:

1. (Currently Amended) A computer-implemented method, comprising:
 - downloading, at a remote site, an application to run as a remote application on a virtual machine located on the remote site, which virtual machine is run in a debug mode to enable the application to attach to the virtual machine;
 - modifying the remote application at the remote site to generate a modified remote application;
 - requesting, from the remote site, a local site to run a debugging system on the modified remote application running on the remote site;
 - running the debugging system on the local site, the local site being separated from the remote site by at least one firewall;
 - establishing a communication link between a first router located on the local site and a second router located on the remote site;
 - using the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;
 - receiving a timestamp indicating when the remote application was last modified at the remote site;
 - determining, based on receiving the timestamp, whether a local copy of the remote application is present on the local site, wherein the local copy is provided as source code;
 - retrieving, if it is determined that the local copy is not present on the local site, the local copy from the remote site;
 - determining, based on receiving the timestamp, whether more than one local copy is stored on the local site;
 - selecting a matching local copy from the more than one local copy based on the received timestamp;
 - determining whether the local copy is up to date with respect to the remote application

based on the received timestamp;

if it is determined that the local copy is not up to date:

determining delta information that identifies differences between the local copy as originally downloaded with the remote application and the modified remote application,

retrieving the delta information from the remote site,

using the retrieved delta information to alter the local copy to match the modified remote application and to provide an altered copy,

establishing an identification mark for the altered copy, which identification mark is stored in the altered copy, and

loading the altered copy as an ~~retrieved, matched or~~ altered local copy into the debugging system; and

debugging, using the communication link between the first and second routers, the ~~loaded~~ altered local copy of the remote application at the local site, further comprising:

sending commands from the local site to the remote application between the first and second routers, and

receiving, via the first and second routers, run-time data and state information about the remote application at the local site based on sending the commands.

2. (Original) The method of claim 1, wherein the remote application is a component of a larger application that is running on the virtual machine.
3. (Original) The method of claim 1, wherein the virtual machine is a Java virtual machine.
4. (Original) The method of claim 1, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.
5. (Cancelled)
6. (Cancelled)

7. (Currently Amended) A computer program product, tangibly embodied in a machine-readable storage device, the computer program product being operable to cause data processing apparatus to:

download, at a remote site, an application to run as a remote application on a virtual machine located on the remote site, which virtual machine is run in a debug mode to enable the application to attach to the virtual machine;

modify the remote application at the remote site to generate a modified remote application;

request, from the remote site, a local site to run a debugging system on the modified remote application running on the remote site;

run the debugging system on the local site, the local site being separated from the remote site by at least one firewall;

establish a communication link between a first router located on the local site and a second router located on the remote site;

use the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;

receive a timestamp indicating when the remote application was last modified at the remote site;

determine, based on receiving the timestamp, whether a local copy of the remote application is present on the local site, wherein the local copy is provided as source code;

retrieve, if it is determined that the local copy is not present on the local site, the local copy from the remote site;

determine, based on receiving the timestamp, whether more than one local copy is stored on the local site;

select a matching local copy from the more than one local copy based on the received timestamp;

determine whether the local copy is up to date with respect to the remote application based on the received timestamp;

if it is determined that the local copy is not up to date:

determine delta information that identifies differences between the local copy as original downloaded with the remote application and the modified remote application,

retrieve the delta information from the remote site,

use the retrieved delta information to alter the local copy to match the modified remote application and to provide an altered copy,

establish an identification mark for the altered copy, which identification mark is stored in the altered copy, and

load the altered copy as an retrieved, matched or altered local copy into the debugging system; and

debug, using the communication link between the first and second routers, the loaded altered local copy of the remote application at the local site, further comprising:

sending commands from the local site to the remote application between the first and second routers, and

receiving, via the first and second routers, run-time data and state information about the remote application at the local site based on sending the commands.

8. (Original) The product of claim 7, wherein the remote application is a component of a larger application that is running on the virtual machine.

9. (Original) The product of claim 7, wherein the virtual machine is a Java virtual machine.

10. (Original) The product of claim 7, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.

11. (Previously Presented) The product of claim 7, wherein at least one of the first and second routers is an SAProuter.

12. (Cancelled)

13. (Cancelled)

14. (Currently Amended) A data processing system, comprising:

a first computer system at a remote site, comprising:

means for downloading, at the remote site, an application to run as a remote application on a virtual machine located on the remote site, which virtual machine is run in a debug mode to enable the application to attach to the virtual machine;

means for modifying the remote application at the remote site to generate a modified remote application;

means for requesting, from the remote site, a local site to run a debugging system on the modified remote application running on the remote site; and

a second computer system at the local site, comprising:

means for running the debugging system on the local site, the local site being separated from the remote site by at least one firewall; and

wherein the data processing system further comprises:

means for establishing a communication link between a first router located on the local site and a second router located on the remote site;

means for using the communication link between the first and second routers to establish communication between the debugging system and the virtual machine;

means for receiving a timestamp indicating when the remote application was last modified at the remote site;

means for determining, based on receiving the timestamp, whether a local copy of the remote application is present on the local site, wherein the local copy is provided as source code;

means for retrieving, if it is determined that the local copy is not present on the local site, the local copy from the remote site;

means for determining, based on receiving the timestamp, whether more than one local copy is stored on the local site;

means for selecting a matching local copy from the more than one local copy based on the received timestamp;

means for determining whether the local copy is up to date with respect to the

remote application based on the received timestamp;

means for determining, if it is determined that the local copy is not up to date, delta information that identifies differences between the local copy as original downloaded with the remote application and the modified remote application,

means for retrieving, if it is determined that the local copy is not up to date, the delta information from the remote site,

means for using, if it is determined that the local copy is not up to date, the retrieved delta information to alter the local copy to match the modified remote application and to provide an altered copy,

means for establishing an identification mark for the altered copy, which identification mark is stored in the altered copy, and

means for loading the altered copy as an retrieved, ~~matched or~~ altered local copy into the debugging system; and

means for debugging, using the communication link between the first and second routers, the ~~loaded~~ altered local copy of the remote application at the local site, further comprising:

means for sending commands from the local site to the remote application between the first and second routers, and

means for receiving, via the first and second routers, run-time data and state information about the remote application at the local site based on sending the commands.

15. (Original) The system of claim 14, wherein the remote application is a component of a larger application that is running on the virtual machine.

16. (Original) The system of claim 14, wherein the virtual machine is a Java virtual machine.

17. (Original) The system of claim 14, wherein the at least one firewall comprises a first firewall protecting the local site and a second firewall protecting the remote site.

18. (Previously Presented) The system of claim 14, wherein at least one of the first and second routers is an SAProuter.
19. (Cancelled)
20. (Cancelled)
21. (Previously Presented) The method of claim 1, further comprising changing a run mode of the modified remote application from a normal mode to a debugging mode.
22. (Previously Presented) The method of claim 1, wherein communication between the debugging system and the virtual machine is established with the modified remote application remaining in a normal mode, and not a debugging mode.
23. (Previously Presented) The method of claim 1, wherein the communication link is established using shared memory.